Frequency-aware Decomposition Learning for Sensorless Wrench Forecasting on a Vibration-rich Hydraulic Manipulator

Hyeonbeen Lee, Min-Jae Jung, Tae-Kyeong Yeu, Jong-Boo Han, Daegil Park, and Jin-Gyun Kim

***Abstract*—Force and torque (F/T) sensing is critical for robot-environment interaction, but physical F/T sensors impose constraints in size, cost, and fragility. To mitigate this, recent studies have estimated force/wrench sensorlessly from robot internal states. While existing methods generally target relatively slow interactions, tasks involving rapid interactions, such as grinding, can induce task-critical high-frequency vibrations, and estimation in such robotic settings remains underexplored. To address this gap, we propose a Frequency-aware Decomposition Network (FDN) for short-term forecasting of vibration-rich wrench from proprioceptive history. FDN predicts spectrally decomposed wrench with asymmetric deterministic and probabilistic heads, modeling the high-frequency residual as a learned conditional distribution. It further incorporates frequency-awareness to adaptively enhance input spectra with learned filtering and impose a frequency-band prior on the outputs. We pretrain FDN on a large-scale open-source robot dataset and transfer the learned proprioception-to-wrench representation to the downstream. On real-world grinding excavation data from a 6-DoF hydraulic manipulator and under a delayed estimation setting, FDN outperforms baseline estimators and forecasters in the high-frequency band and remains competitive in the low-frequency band. Transfer learning provides additional gains, suggesting the potential of large-scale pretraining and transfer learning for robotic wrench estimation. Code and data are available at GitHub.**

***Index Terms*—Force and torque estimation, contact-rich manipulation, transfer learning, hydraulic manipulators, industrial robotics**

I. Introduction

FORCE and torque (F/T) sensing is critical for robotic applications as it provides direct information about contact, and has been particularly highlighted in minimally invasive surgery [1], haptic feedback in teleoperation systems [2], force-based control [3], and force-informed robot learning [4]. However, measuring these quantities often requires the installation of physical sensors, which introduces bottlenecks including their size, weight, fragility, and cost, thereby limiting their widespread deployment [5].

To address hardware limitations, sensorless approaches for robotic force or wrench estimation have been actively studied. Widely used model-based methods leverage inverse robot dynamics or state observers, which rely on identified mathematical models of system dynamics [6]. These methods are physically interpretable and theoretically grounded, but are generally less adaptable to varying environments and systems. Challenges also remain in parameter identification, dynamics modeling, and numerical stability [7], [8].

Along with recent advances in machine learning, data-driven approaches for robotic force or wrench estimation have also received increasing attention. Typically based on neural networks and regression models, they capture underlying dynamics directly from data without the need for explicit mathematical models, which offers enhanced modeling flexibility [6], [9]–[12].

However, most existing data-driven methods for robotic force or wrench estimation only partially incorporate advances in modern machine learning. In particular, recent deep time-series forecasting models improve sequential modeling through decomposition-, frequency-, and patch-based architectures [13], [14], which may be suitable for capturing the temporal structure of output wrench trajectories. They also have potential for delay-compensated prediction beyond conventional instantaneous estimation. Moreover, large-scale pretraining and transfer learning have demonstrated strong potential for improving generalization across tasks, embodiments, and environments in robot learning [15]. In parallel, high-quality multimodal robot datasets including wrench measurements have become available [16]. These developments motivate the exploration of large-scale pretraining for data-driven wrench estimation. Nevertheless, the integration of the aforementioned advances into the literature remains underexplored.

Another underexplored challenge is the estimation of high-frequency wrench components. Most previous works focus on smooth wrench signals arising from relatively slow interactions such as grasping [6], [9]–[12]. However, tasks involving more rapid interactions, such as robotic grinding [17], milling [18], and polishing [19], which are prevalent in industrial and surgical robotics, can generate substantial task-relevant high-frequency vibrations. Additional vibrations may also arise from the actuation mechanism itself, for instance, through pressure fluctuations in hydraulic actuation systems [20]. These signals are often difficult to estimate because of the low-frequency bias and overfitting of neural networks [21], [22], as well as model mismatch, derivative noise, and observer lag in model-based estimators [6], [17], [23]. Accordingly, the effectiveness of sensorless wrench estimation in contact- and vibration-rich settings remains insufficiently validated. Related studies on machining force prediction address similar phenomena [24], [25], but they focus on more periodic and

This research was supported by a grant from Endowment Project of "Development of Core Technologies for Operation of Marine Robots based on Cyber-Physical System" funded by Korea Research Institute of Ships and Ocean Engineering (PES5200) (*Corresponding author: Jin-Gyun Kim*).

H. Lee, M.J. Jung, and J.G. Kim are with the Department of Mechanical Engineering, Kyung Hee University, South Korea (e-mail: {lhbsharp; jmhahh; jingyun.kim}@khu.ac.kr).

T.K. Yeu, J.B. Han, and D. Park are with the Korea Research Institute of Ships and Ocean Engineering (KRISO), South Korea (e-mail: {yeutk; jbhan; daegilpark}@kriso.re.kr)

homogeneous oscillations than the transient and unstructured wrench fluctuations encountered in robotic interactions [17]–[19].

Altogether, these gaps motivate a deliberate integration of modern machine learning methods for contact- and vibration-rich wrench estimation in robotics. To this end, we propose a Frequency-aware Decomposition Network (FDN) that incorporates decomposition-based probabilistic modeling, frequency-awareness, and large-scale proprioception-to-wrench pretraining for sensorless short-term wrench forecasting. We validate the proposed framework on real-world grinding excavation with a 6-DoF hydraulic manipulator, where the target wrench exhibits substantial high-frequency vibrations arising from rapid contact transients. Under this setting, we compare the proposed framework against baselines from both robotic wrench estimation and time-series forecasting, with particular attention to band-specific performance in the low- and high-frequency ranges under delayed estimation. Ablation studies and transfer analyses further examine the effectiveness and behavior of the proposed design choices.

II. Preliminaries

A. Robot dynamics and wrench estimation

The dynamic model of a robot manipulator in joint space $\mathbb{R}^n$ is generally expressed as:

$$\mathrm{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathrm{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathrm{G}(\boldsymbol{q}) + \mathrm{F}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \mathrm{J}(\boldsymbol{q})^T\mathbf{W} \quad (1)$$

where $\boldsymbol{q} \in \mathbb{R}^n, \dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ represent joint position, velocity, and acceleration vectors, respectively. $\mathrm{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ denotes a positive definite inertia matrix, $\mathrm{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal matrix, $\mathrm{G}(\boldsymbol{q}), \mathrm{F}(\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ denote gravity and friction terms. $\boldsymbol{\tau} \in \mathbb{R}^n$ is joint actuator torque, $\mathrm{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ is the robot Jacobian, and $\mathbf{W} = [\mathbf{f};\mathbf{m}] \in \mathbb{R}^6$ is Cartesian space wrench induced by robot-environment interaction. To solve the second-order dynamics, we define a state vector $\boldsymbol{y} = [\boldsymbol{q},\dot{\boldsymbol{q}}]^T \in \mathbb{R}^{2n}$ and rewrite Eq. 1 as a first-order system:

$$\dot{\boldsymbol{y}} = h(\boldsymbol{y},\boldsymbol{\tau},\mathbf{W}) \quad (2)$$

where $\dot{\boldsymbol{y}} = [\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}]^T$ and

$$\ddot{\boldsymbol{q}} = \mathrm{M}^{-1}(\boldsymbol{q})\left(\boldsymbol{\tau} + \mathrm{J}(\boldsymbol{q})^T\mathbf{W} - \mathrm{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \mathrm{G}(\boldsymbol{q}) - \mathrm{F}(\dot{\boldsymbol{q}})\right) \quad (3)$$

Then, the solution of Eq. 2 can be obtained via recursive numerical integration:

$$\boldsymbol{y}_{t+1} = \Phi_{\Delta t}(\boldsymbol{y}_t, \dot{\boldsymbol{y}}_t) \quad (4)$$

where $\Phi_{\Delta t}$ denotes a numerical integrator with a time step size Δt.

Traditionally, the inverse dynamics model of Eq. 1 is favored for wrench or force estimation:

$$\begin{aligned}\mathbf{W} &= \mathrm{J}^{-T}(\boldsymbol{q})\big(\mathrm{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathrm{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathrm{G}(\boldsymbol{q}) + \mathrm{F}(\dot{\boldsymbol{q}}) - \boldsymbol{\tau}\big) \\ &= \psi(\boldsymbol{y},\dot{\boldsymbol{y}},\boldsymbol{\tau})\end{aligned} \quad (5)$$

In contrast, data-driven models such as neural networks do not require an explicit mathematical model and can capture the underlying dynamics of $\mathbf{W}$ directly from the data [6], [9], [10], [12].

B. Wrench forecasting model

Beyond instantaneous estimation of $\mathbf{W}_t$ based on Eq. 1, we extend the formulation to forecasting a future wrench sequence $[\mathbf{W}_{t+1}, \mathbf{W}_{t+2}, \cdots]$, which allows the sequential modeling of the wrench trajectory in a predictive manner. By combining Eq. 2 and 4 into Eq. 5, we obtain the one-step dependence:

$$\mathbf{W}_{t+1} = \psi\big(\Phi_{\Delta t}\big(\boldsymbol{y}_t, h(\boldsymbol{y}_t,\boldsymbol{\tau}_t,\mathbf{W}_t)\big), \dot{\boldsymbol{y}}_{t+1}, \boldsymbol{\tau}_{t+1}\big) \quad (6)$$

which implies temporal dependence between one-step-ahead wrench $\mathbf{W}_{t+1}$ and current joint states $\boldsymbol{y}_t = [\boldsymbol{q}_t,\dot{\boldsymbol{q}}_t]^T$, actuator torque $\boldsymbol{\tau}_t$, and wrench $\mathbf{W}_t$. For a sufficiently small Δt and locally smooth states, we can adopt local approximations $\dot{\boldsymbol{y}}_{t+1} \approx \dot{\boldsymbol{y}}_t$ and $\boldsymbol{\tau}_{t+1} \approx \boldsymbol{\tau}_t$ and rewrite Eq. 6 as:

$$\mathbf{W}_{t+1} = \mathcal{T}_{\Delta t}(\boldsymbol{q}_t,\dot{\boldsymbol{q}}_t,\ddot{\boldsymbol{q}}_t,\boldsymbol{\tau}_t,\mathbf{W}_t) + \boldsymbol{e}_{t+1} \quad (7)$$

where we define an approximate one-step transition $\mathcal{T}_{\Delta t}$ and an approximation error term $\boldsymbol{e}_{t+1}$. Letting $\mathcal{X}_t = [\boldsymbol{q}_t,\dot{\boldsymbol{q}}_t,\ddot{\boldsymbol{q}}_t,\boldsymbol{\tau}_t]$ and rolling out $\mathcal{T}_{\Delta t}$ yields:

$$\mathbf{W}_{t+k} \approx \mathcal{T}_{\Delta t}(\mathcal{X}_{t+k-1}, \mathbf{W}_{t+k-1}), \quad k = 1,\cdots,T \quad (8)$$

which suggests that $\mathbf{W}_{t+1:t+T} = [\mathbf{W}_{t+1},\cdots,\mathbf{W}_{t+T}]$ depends on the trajectories of recursively propagated motion, actuator torque, and wrench up to time $t+T-1$.

The approximate recursive dependency in Eq. 8 motivates short-term forecasting, but the rollout is not feasible due to partial observability. Specifically, we presume a situation where (i) the wrench is not measurable and (ii) the states are observed up to the present time t. Alternatively, we can model a conditional distribution of the T-step future wrench sequence given a finite L-step history of observable states $\boldsymbol{x}$:

$$\mathbf{W}_{t+1:t_f} \sim \mathcal{P}(\cdot \mid \boldsymbol{x}_{t_h:t}), \quad \boldsymbol{x}_t = [\boldsymbol{q}_t,\dot{\boldsymbol{q}}_t,\ddot{\boldsymbol{q}}_t,\boldsymbol{u}_t] \quad (9)$$

where $t_h = t - L + 1$ is history start index and $t_f = t + T$ is future end index. Here, we replace $\boldsymbol{\tau}$ with an observable actuation signal $\boldsymbol{u}$, such as joint differential hydraulic pressure, motor torque, or motor current. We then approximate $\mathcal{P}$ with a model $\mathbf{M}_\theta$:

$$\hat{\mathbf{W}}_{t+1:t_f} \sim \mathbf{M}_\theta(\cdot \mid \boldsymbol{x}_{t_h:t}) \quad (10)$$

θ denotes learnable parameters of the model. To bridge the discrepancy between Eq. 8 and 10 where—(i) $\boldsymbol{\tau}$ is replaced with $\boldsymbol{u}$, (ii) $\mathbf{W}$ is removed, and (iii) time steps are observable up to t only—we assume that (i) $\boldsymbol{u}$ provides sufficient information about $\boldsymbol{\tau}$, (ii) motivated by Eq. 1, a finite history of robot states retains information about recent wrench effects, and (iii) we focus on a short-term forecasting horizon T, where the system dynamics typically evolve smoothly and a finite history $\boldsymbol{x}_{t_h:t}$ can serve as an informative summary of recent interaction trends. Then, we can let the model $\mathbf{M}_\theta$ capture the underlying correlations from data while absorbing these modeling assumptions.

III. Frequency-aware Decomposition Network

A. Learning from spectral decomposition

The proposed FDN model is illustrated in Fig. 1. To address the challenge of learning the high-frequency dynamics in our

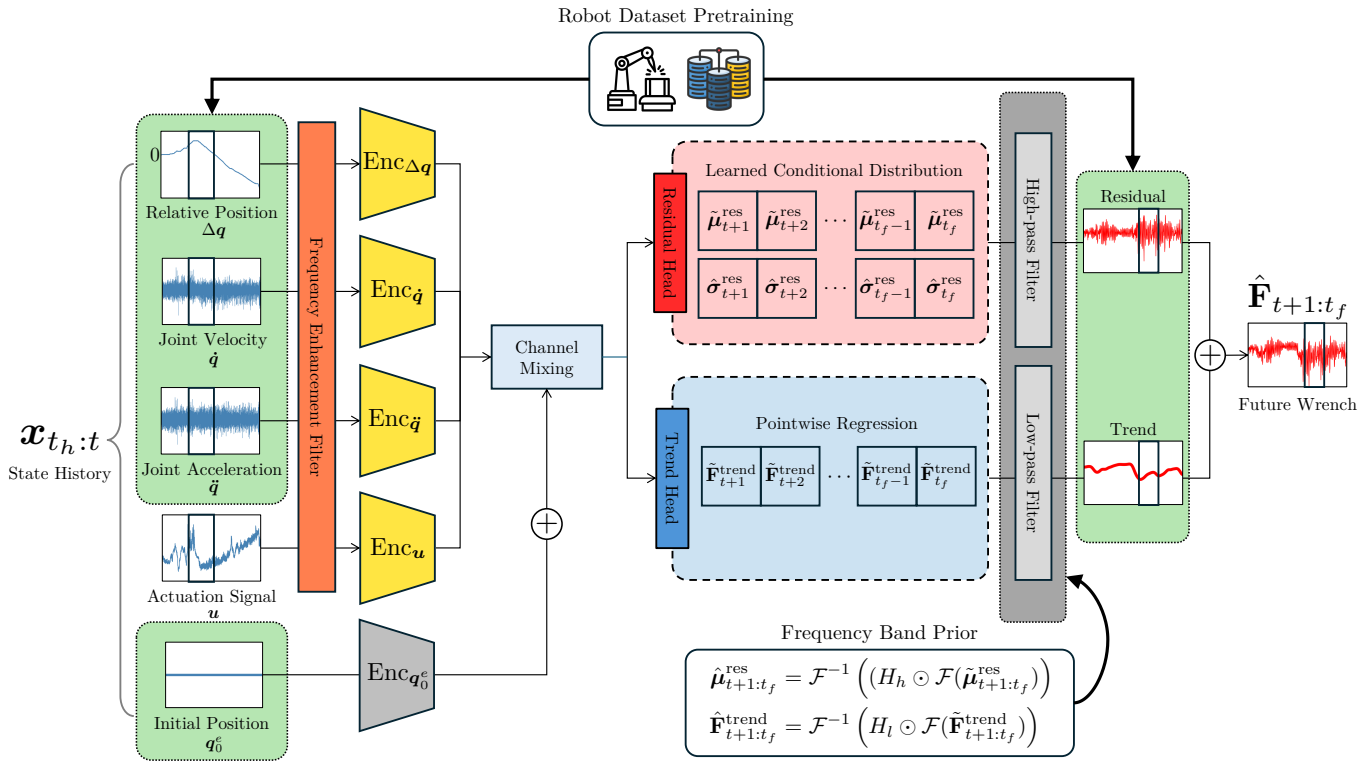


Fig. 1. Illustration of the proposed Frequency-aware Decomposition Network (FDN).

wrench forecasting, we employ asymmetric modeling for the low- and high-frequency bands of the wrench signal. To this end, we decompose $\mathbf{W}$ into trend and residual over each T-step horizon, and then let our model $\mathbf{M}_\theta$ forecast a tuple of T-step decompositions. Here, $\mathbf{W}$ denotes episode-level denoised wrench with a cutoff frequency f_c^{dn}. For each prediction time t, the ground-truth trend and residual sequences are defined using spectral decomposition as:

$$\begin{aligned}\mathbf{W}^{\text{trend}}_{t+1:t_f} &= \text{FPF}_{\text{low}}(\mathbf{W}_{t+1:t_f}) \\ \mathbf{W}^{\text{res}}_{t+1:t_f} &= \mathbf{W}_{t+1:t_f} - \mathbf{W}^{\text{trend}}_{t+1:t_f}\end{aligned} \tag{11}$$

where

$$\text{FPF}_{\text{low}}(\cdot) = \mathcal{F}^{-1}\left(H_l(f; f_c) \odot \mathcal{F}(\cdot)\right) \tag{12}$$

is a differentiable non-recursive low-pass filter where

$$H_l(f; f_c) = 1/\sqrt{1 + (f/f_c)^{2r}}, \quad f \in [0, f_{\text{Nyq}}] \tag{13}$$

is a real-valued low-pass amplitude response with cutoff frequency f_c [26]. f_{Nyq} is the Nyquist frequency. $\mathcal{F}, \mathcal{F}^{-1}$ are the Fast Fourier Transform (FFT) and its inverse, and $\odot$ is an elementwise multiplication operator.

Then, the FDN model $\mathbf{M}_\theta$ learns to forecast wrench decompositions given $\boldsymbol{x}_{t_h:t}$ as:

$$\{\hat{\mathbf{W}}^{\text{trend}}_{t+1:t_f}, \hat{\mathbf{W}}^{\text{res}}_{t+1:t_f}\} \sim \mathbf{M}_\theta(\cdot \mid \boldsymbol{x}_{t_h:t}) \tag{14}$$

The final forecast output $\hat{\mathbf{W}}_{t+1:t_f}$ is obtained by summing the trend and residual predictions:

$$\hat{\mathbf{W}}_{t+1:t_f} = \hat{\mathbf{W}}^{\text{trend}}_{t+1:t_f} + \hat{\mathbf{W}}^{\text{res}}_{t+1:t_f} \tag{15}$$

To reduce boundary artifacts in Eq. 12, we apply both-sided reflection padding before FFT and center-crop the inverse transformed sequence. We set $r = 8$.

B. Modality-specific encoders

To reduce the model's sensitivity to episode-dependent initial positions, we redefine the input vector $\boldsymbol{x}_t$ using the relative joint positions $\Delta\boldsymbol{q}$ and episode-dependent initial positions $\boldsymbol{q}_0^e$:

$$\boldsymbol{x}_t = [\Delta\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \ddot{\boldsymbol{q}}_t, \boldsymbol{u}_t, \boldsymbol{q}_0^e] \in \mathbb{R}^{5n} \tag{16}$$

for a n-DoF robot, where

$$\Delta\boldsymbol{q}_t = \boldsymbol{q}_t - \boldsymbol{q}_0^e \tag{17}$$

and $\boldsymbol{q}_0^e$ is the episode initial position.

Given an input history $\boldsymbol{x}_{t_h:t} \in \mathbb{R}^{5n \times L}$, our model computes a sequence representation using four modality-specific PatchTST [14] encoders $\text{Enc}_{\Delta\boldsymbol{q}}, \text{Enc}_{\dot{\boldsymbol{q}}}, \text{Enc}_{\ddot{\boldsymbol{q}}}$, and $\text{Enc}_{\boldsymbol{u}}$ for time-varying modalities and a MLP $\text{Enc}_{\boldsymbol{q}_0^e}$ for episode-varying initial positions. For the time-varying subset $\boldsymbol{x}^{\delta}_{t_h:t} = [\Delta\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \ddot{\boldsymbol{q}}_t, \boldsymbol{u}_t] \in \mathbb{R}^{4n \times L}$, we first enhance them in the frequency domain as described in Section III-D, then patch-embed $\text{FEF}(\boldsymbol{x}^{\delta}_{t_h:t})$ to a latent dimension D with patch length P and stride $S = P$ using modality-specific embedding layers. Subsequently, we obtain representations $\{z_{\Delta\boldsymbol{q}}, z_{\dot{\boldsymbol{q}}}, z_{\ddot{\boldsymbol{q}}}, z_{\boldsymbol{u}}\} \in \mathbb{R}^{n \times N \times D}$ from the embeddings using modality-specific Transformer [27] encoders, where $N = \lfloor (L - P)/S \rfloor + 2$ is the number of patches. For $\boldsymbol{q}_0^e$, we compute D-dimensional representation $z_{\boldsymbol{q}_0^e}$ with $\text{Enc}_{\boldsymbol{q}_0^e}$, broadcast them across patches and channels, and add it to $z_{\Delta\boldsymbol{q}}$. Then, we concatenate

the representations along the channel dimension, yielding $z' = [z_{\Delta \boldsymbol{q}} + z_{\boldsymbol{q}_0^e}, z_{\dot{\boldsymbol{q}}}, z_{\ddot{\boldsymbol{q}}}, z_{\boldsymbol{u}}] \in \mathbb{R}^{4n \times N \times D}$. Finally, we apply a channel-mixing linear projection to z' to map the channel dimension from $4n$ to the 6 wrench channels, and obtain the final representation $z \in \mathbb{R}^{6 \times N \times D}$.

a) Modified reversible instance normalization: The original PatchTST applies reversible instance normalization (RevIN) [28] before patch embedding, which normalizes each input channel and denormalizes the model outputs using sample-wise statistics. This technique assumes identical input and output channels, which differs from our setting. Here, we take advantage of the RevIN by normalizing $\boldsymbol{x}^{\delta}_{t_h:t}$ before frequency enhancement, and applying the inverse transform to the representations $\{z_{\Delta \boldsymbol{q}}, z_{\dot{\boldsymbol{q}}}, z_{\ddot{\boldsymbol{q}}}, z_{\boldsymbol{u}}\}$.

C. Asymmetric forecasting heads

From the final representation z, the model forecasts trend $\tilde{\mathbf{W}}^{\text{trend}}$ and residual distribution parameters in the short-term future using separate linear heads. We first flatten the patch and latent dimensions of z:

$$\bar{z} = \text{flatten}_{N,D}(z) \in \mathbb{R}^{6 \times ND} \tag{18}$$

Then, for the trend, we linearly project the flattened dimension ND to T as:

$$\tilde{\mathbf{W}}^{\text{trend}}_{t+1:t_f} = \bar{z} W_{\text{trend}} + b_{\text{trend}} \tag{19}$$

where $W_{\text{trend}} \in \mathbb{R}^{ND \times T}$, $b_{\text{trend}} \in \mathbb{R}^{T}$, and $\tilde{\mathbf{W}}^{\text{trend}}_{t+1:t_f} \in \mathbb{R}^{6 \times T}$.

For the residual, we model it as a step- and channel-wise Gaussian distribution in the T-step horizon, and predict conditional distribution parameters as:

$$\begin{aligned} \tilde{\boldsymbol{\mu}}^{\text{res}}_{t+1:t_f} &= \bar{z} W_{\mu} + b_{\mu} \\ \hat{\mathbf{v}}^{\text{res}}_{t+1:t_f} &= \bar{z} W_{v} + b_{v} \end{aligned} \tag{20}$$

where $W_{\mu}, W_{v} \in \mathbb{R}^{ND \times T}$ and $b_{\mu}, b_{v} \in \mathbb{R}^{T}$. Outputs $\tilde{\boldsymbol{\mu}}^{\text{res}}_{t+1:t_f}, \hat{\mathbf{v}}^{\text{res}}_{t+1:t_f} \in \mathbb{R}^{6} \times T$ denote predicted mean and log variances, and $\hat{\boldsymbol{\sigma}}^{\text{res}}_{t+1:t_f} = \exp(\hat{\mathbf{v}}^{\text{res}}_{t+1:t_f}/2)$. Modeling the residual as a probabilistic distribution can efficiently parameterize volatile high-frequency amplitudes with the learned distribution, which is the key design of the FDN.

D. Frequency-awareness

We incorporate frequency-aware layers in FDN to enhance inputs and refine outputs in the frequency domain, as in Fig. 2. To utilize our frequency band prior obtained from spectral decomposition and denoising, we filter the model predictions $\tilde{\mathbf{W}}^{\text{trend}}_{t+1:t_f}$ and $\tilde{\boldsymbol{\mu}}^{\text{res}}_{t+1:t_f}$ as:

$$\begin{aligned} \hat{\mathbf{W}}^{\text{trend}}_{t+1:t_f} &= \text{FPF}_{\text{low}}(\tilde{\mathbf{W}}^{\text{trend}}_{t+1:t_f}) \\ \hat{\boldsymbol{\mu}}^{\text{res}}_{t+1:t_f} &= \text{FPF}_{\text{high}}(\tilde{\boldsymbol{\mu}}^{\text{res}}_{t+1:t_f}) \end{aligned} \tag{21}$$

where

$$\text{FPF}_{\text{high}}(\cdot) = \mathcal{F}^{-1}\left(H_h(f; f_c, f_c^{\text{dn}}) \odot \mathcal{F}(\cdot)\right) \tag{22}$$

is a denoising high-pass filter and

$$H_h(f; f_c, f_c^{\text{dn}}) = \big(1 - H_l(f; f_c)\big) H_l(f; f_c^{\text{dn}}) \tag{23}$$

is a band-pass response between f_c and f_c^{dn} ($f_c \le f_c^{\text{dn}}$). Then, we sample the residual at each step independently across time as:

$$\hat{\mathbf{W}}^{\text{res}}_{t+k} = \hat{\boldsymbol{\mu}}^{\text{res}}_{t+k} + \epsilon \odot \exp\left(\frac{\hat{\mathbf{v}}^{\text{res}}_{t+k}}{2}\right) \quad \forall k = 1, \cdots, T \tag{24}$$

to generate a residual sequence $\hat{\mathbf{W}}^{\text{res}}_{t+1:t_f} \in \mathbb{R}^{6 \times T}$. Here, $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$ is a white noise vector in $\mathbb{R}^6$. We use the filtered outputs to compute the loss in Eq. 31 since the filtering operations in Eq. 21 are differentiable. Here, $\hat{\mathbf{W}}^{\text{res}}_{t+1:t_f}$ can be further filtered using FPF_{high} for sample-level refinement, but we omit this design to keep the sampling path simple.

To additionally mitigate high-frequency learning by adaptively scaling input frequency components, we apply a learnable frequency enhancement filter to $\boldsymbol{x}^{\delta}_{t_h:t}$ before patch embedding. We define a learnable filter as:

$$f(\boldsymbol{X}) = \boldsymbol{X} \odot \text{softplus}(W_f) \tag{25}$$

where $\boldsymbol{X} = \mathcal{F}(\boldsymbol{x}^{\delta}_{t_h:t}) \in \mathbb{C}^{4n \times (\lfloor L/2 \rfloor + 1)}$ is a Fourier transform of $\boldsymbol{x}^{\delta}_{t_h:t}$ and $W_f \in \mathbb{R}^{4n \times (\lfloor L/2 \rfloor + 1)}$ is a learnable real-valued weight. The frequency enhancement is executed by a mixture-of-experts (MoE) over M learnable filters $f_m(\cdot)$ for $m = 1, \cdots, M$ with learned expert-varying weights. Here, each f_m has its own filter weight $W_{f,m}$. Let a $\alpha_m \in \mathbb{R}$ denote an m-th expert weight. The weight vector $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_M] \in \mathbb{R}^M$ is computed as:

$$\boldsymbol{\alpha} = \text{softmax}_{\text{e}}\left(\phi(\boldsymbol{x}^{\delta}_{t_h:t})\right) \tag{26}$$

$$\phi(\boldsymbol{x}^{\delta}_{t_h:t}) = \left(\text{flatten}_{4n,L}(\boldsymbol{x}^{\delta}_{t_h:t})\right)^T W_p \tag{27}$$

ϕ is a linear gating layer with a weight $W_p \in \mathbb{R}^{4nL \times M}$ to produce M logits, and the $\text{softmax}_{\text{e}}$ denotes softmax function applied over the expert dimension. The frequency enhancement filter $\text{FEF}(\cdot)$ is then defined as:

$$\text{FEF}(\boldsymbol{x}^{\delta}_{t_h:t}) = \sum_{m=1}^{M} \alpha_m \mathcal{F}^{-1}(f_m(\boldsymbol{X})) \tag{28}$$

From an input sequence, it produces M frequency-enhanced sequences and computes a weighted sum of them in the time domain with learned softmax weights, resulting in the same dimensionality as $\boldsymbol{x}^{\delta}_{t_h:t}$. We then patch-embed $\text{FEF}(\boldsymbol{x}^{\delta}_{t_h:t})$ to feed each modality encoder. We use $M = 32$ by default.

E. Loss function

We train the trend head by minimizing the mean-squared error of $\hat{\mathbf{W}}^{\text{trend}}_{t+1:t_f}$:

$$\mathcal{L}_{\text{trend}} = \frac{1}{6T} \sum_{i=1}^{6} \sum_{j=1}^{T} \left(\mathbf{W}^{\text{trend}}_{i,t+j} - \hat{\mathbf{W}}^{\text{trend}}_{i,t+j}\right)^2 \tag{29}$$

where i and $t+j$ denote channel and time indices. The residual head is trained by minimizing the negative log-likelihood of the ground-truth residual sequence $\mathbf{W}^{\text{res}}_{t+1:t_f}$:

$$\mathcal{L}_{\text{res}} = \frac{1}{6T} \sum_{i=1}^{6} \sum_{j=1}^{T} \frac{1}{2} \left(\frac{(\mathbf{W}^{\text{res}}_{i,t+j} - \hat{\boldsymbol{\mu}}^{\text{res}}_{i,t+j})^2}{\exp(\hat{\mathbf{v}}^{\text{res}}_{i,t+j})} + \hat{\mathbf{v}}^{\text{res}}_{i,t+j} \right) \tag{30}$$

The total loss $\mathcal{L}$ is the sum of the trend and residual losses:

$$\mathcal{L} = \mathcal{L}_{\text{trend}} + \mathcal{L}_{\text{res}} \tag{31}$$

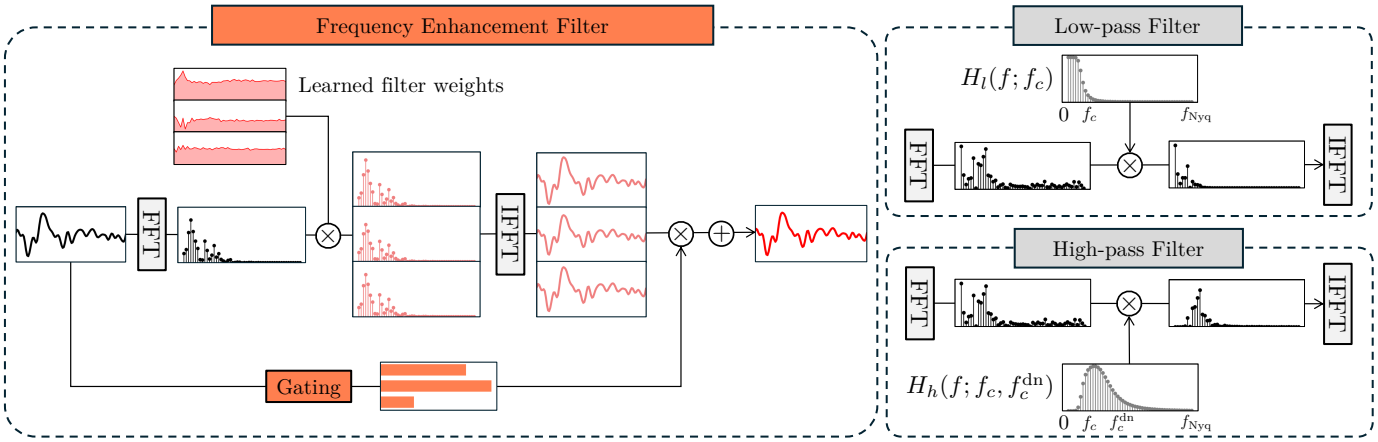


Fig. 2. Detailed visualization of frequency-aware layers. FFT and IFFT denote the Fast Fourier Transform and its inverse. (Left) A learnable frequency enhancement filter (FEF) layer. (Right) Low-pass and denoising high-pass filters ($\text{FPF}_{\text{low}}, \text{FPF}_{\text{high}}$) for imposing the frequency band prior.

F. Proprioception-to-Wrench pretraining

To further enhance generalization, we pretrain the model on RH20T [16], a large-scale open-source robot dataset containing proprioception and wrench data obtained from teleoperated contact-rich manipulations. Specifically, we use the provided $\boldsymbol{q}$ and $\mathbf{W}$ trajectories, and numerical differentiations of the provided $\boldsymbol{q}$ to learn proprioception-to-wrench representations without modifying the defined input, output, and loss settings. Since the dataset covers both 6- and 7-DoF robots, we set $n = 7$ to construct a 35-dimensional input vector and set the 7th-joint channels $\Delta q_7, \dot{q}_7, \ddot{q}_7$, and $q^e_{0,7}$ to zero in $\boldsymbol{x}_{t_h:t}$ and $\text{FEF}(\boldsymbol{x}^{\delta}_{t_h:t})$ for 6-DoF samples. To avoid negative transfer from the modality gap between motor-based actuation and downstream hydraulic actuation in Section IV-A, we mask the actuation signal $\boldsymbol{u}$ with zero in both $\boldsymbol{x}^{\delta}_{t_h:t}$ and $\text{FEF}(\boldsymbol{x}^{\delta}_{t_h:t})$, and zero out the corresponding encoder representation $z_{\boldsymbol{u}}$ to ignore the actuation signal $\boldsymbol{u}$ while pretraining. Accordingly, parameters of the $\text{Enc}_{\boldsymbol{u}}$ and its embedding layer are frozen. We also skip RevIN for the masked channels.

We transfer the learned proprioception-to-wrench representations to the downstream by initializing $\text{Enc}_{\boldsymbol{\Delta q}}, \text{Enc}_{\dot{\boldsymbol{q}}}, \text{Enc}_{\ddot{\boldsymbol{q}}}$ and their embedding layers, and $\text{Enc}_{\boldsymbol{q}^e_0}$ with the pretrained parameters, while initializing the other layers from scratch. In the downstream, we first perform linear probing while freezing the pretrained parameters, and then unfreeze all parameters to fine-tune the model end-to-end.

IV. Experiments

A. Real-world hydraulic manipulation data

We conduct real-world robotic excavations using a 6-DoF KnR HYDRA-UW3 hydraulic manipulator equipped with a grinder end-effector, shown in Fig. 3. To mimic a robotic ground excavation, we teleoperate the manipulator to excavate a fixed gypsum block along the $-\mathbf{x}$ and $-\mathbf{z}$ axes, while maintaining zero translation along the $\mathbf{y}$ axis. The grinder rotates about the $+\mathbf{y}$ direction of the base frame at a desired speed of 100 RPM. During teleoperation, we actuate three intermediate joints (q_2, q_3, q_4) and keep the first (q_1) and the last two joints (q_5, q_6) fixed.

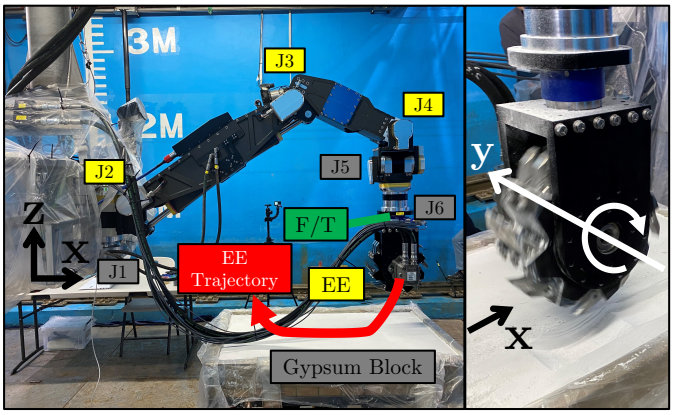


Fig. 3. Illustration of our hydraulic manipulator and grinding excavation. (Left) Overview of our experimental setting. J1 to J6 indicate the joint numbers, and 'EE' denotes the end-effector. Fixed components are colored in grey, while moving components are marked in yellow. (Right) The grinder end-effector rotates about the $+\mathbf{y}$ axis and excavates the block by moving in $-\mathbf{x}$ and $-\mathbf{z}$ directions.

Fig. 4 visualizes the collected trajectories. During excavation, we collect six joint angle trajectories $\boldsymbol{q}$ and wrench from a 6-axis F/T sensor (HBK MCS10) mounted at the wrist. We also collect the joint differential hydraulic pressure $\Delta \boldsymbol{p}$ from pressure sensors installed at each joint, which we use as an actuation signal $\boldsymbol{u}$. The joint states, pressures, and the wrench are sampled at 100 Hz. Note that the wrench measurements are used only for training and evaluating the model.

We collect data over 12 episodes, as summarized in Table I. Here, $N_{(\cdot)}$ denotes the number of samples in each modality. The episodes were collected in two sessions, denoted as 'Soft' and 'Stiff', each including 6 episodes. We excavate *softer blocks rapidly* in the 'Soft' session and *stiffer blocks slowly* in the 'Stiff' session. This setting induced different wrench distributions across sessions, as reflected in the maximum force/torque magnitudes.

1) Selecting cutoff frequencies: We analyze frequency components of the wrench to determine the cutoff frequencies f_c and f_c^{dn}. Fig. 5 shows the average energy spectrum $|\mathcal{F}(\mathbf{W}_{t+1:t_f})|^2$ of undecomposed, 5,000-step wrench windows

TABLE I
OVERVIEW OF THE COLLECTED HYDRAULIC MANIPULATION DATA.

Episode Unit	Dataset -	Duration [s]	In-contact [s]	$N_{\boldsymbol{q},\Delta\boldsymbol{p}}$ -	$N_{\mathbf{W}}$ -	$\max\lvert\mathbf{f}\rvert$ [N]	$\max\lvert\mathbf{m}\rvert$ [Nm]	d_x [mm]	d_z [mm]	$\mathbb{E}[v_x]$ [mm/s]
Soft-1	Test	289.18	214.11	28,861	28,924	$\mathbf{f_x} = 122$	$\mathbf{m_x} = 35$	158.31	11.75	-0.74
Soft-2	Test	149.34	114.15	14,909	14,937	$\mathbf{f_x} = 96$	$\mathbf{m_y} = 27$	154.70	12.58	-1.36
Soft-3	Training	230.33	141.70	22,933	23,035	$\mathbf{f_x} = 155$	$\mathbf{m_y} = 47$	213.84	13.79	-1.50
Soft-4	Training	152.04	108.32	15,121	15,205	$\mathbf{f_z} = 116$	$\mathbf{m_x} = 42$	160.00	12.57	-1.48
Soft-5	Training	176.63	130.39	17,618	17,663	$\mathbf{f_x} = 134$	$\mathbf{m_y} = 41$	206.74	13.23	-1.59
Soft-6	Training	155.16	121.75	15,427	15,517	$\mathbf{f_x} = 111$	$\mathbf{m_y} = 31$	162.21	11.01	-1.33
Stiff-1	Test	418.71	336.52	41,824	41,874	$\mathbf{f_x} = 627$	$\mathbf{m_z} = 193$	235.64	16.27	-0.70
Stiff-2	Training	445.78	375.77	44,579	44,581	$\mathbf{f_x} = 239$	$\mathbf{m_z} = 69$	140.49	13.56	-0.37
Stiff-3	Test	593.85	479.60	59,366	59,381	$\mathbf{f_x} = 364$	$\mathbf{m_z} = 125$	206.57	14.61	-0.43
Stiff-4	Training	396.82	324.99	39,682	39,685	$\mathbf{f_x} = 258$	$\mathbf{m_z} = 78$	157.16	13.36	-0.48
Stiff-5	Training	557.77	457.64	55,456	55,756	$\mathbf{f_x} = 303$	$\mathbf{m_z} = 120$	205.70	12.73	-0.45
Stiff-6	Training	383.12	331.67	38,306	38,293	$\mathbf{f_x} = 260$	$\mathbf{m_z} = 75$	156.05	13.03	-0.47
Total	-	3948.73	3136.60	394,082	394,851	-	-	-	-	-

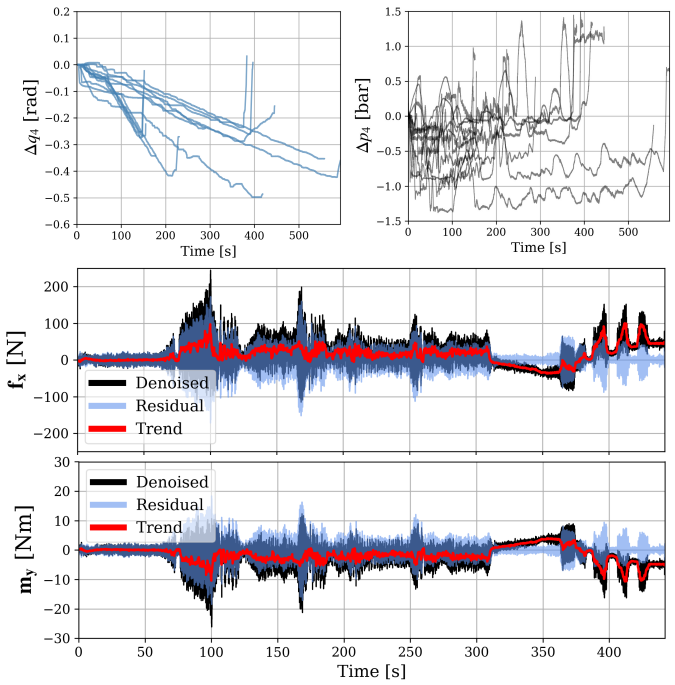


Fig. 4. Visualization of the collected hydraulic dataset. The upper two panels show 12 collected trajectories of Δq_4 and Δp_4. The lower two panels show the decomposed $\mathbf{f_x}$ and $\mathbf{m_y}$ trajectories of the 'Stiff-2' episode with $f_c = 1$ Hz and $f_c^{\mathrm{dn}} = 15$ Hz.

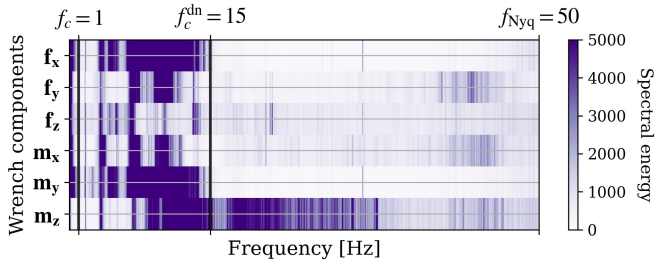


Fig. 5. Energy spectrum of raw wrench windows. Each channel is normalized to zero mean and unit variance before the Fourier transform.

before denoising. We observe that low-frequency energy is concentrated below 1 Hz across all channels, while overall spectral energy is concentrated below 15 Hz. Accordingly, we set $f_c = 1$ Hz and $f_c^{\mathrm{dn}} = 15$ Hz.

B. Data processing

We postprocess the data for training. We denoise the inputs $\boldsymbol{q}$ and $\Delta\boldsymbol{p}$ with cutoff frequency f_c^{dn}, and estimate $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ from the denoised $\boldsymbol{q}$ using a causal Savitzky-Golay filter. We transform the wrench from the sensor frame to the base frame and synchronize the timestamps to those of the joint states using zero-order hold. In addition, we remove episode-wise sensor offsets in $\mathbf{W}$ by subtracting the temporal mean of the static-phase segment for each channel in each episode to ensure near-zero readings in non-contact phases. Then, we denoise $\mathbf{W}$ at the episode level before performing the window-level decomposition in Eq. 11. For the pretraining dataset, we use the same postprocessing pipeline except that we upsample $\boldsymbol{q}$ from 10 Hz to 100 Hz with linear interpolation using timestamps of $\mathbf{W}$, omit sensor offset removal in $\mathbf{W}$, and extract samples with a temporal stride of 10 due to the large number of samples.

The postprocessed data is split at the episode level to construct training and test datasets. To account for session-dependent wrench distributions, we select two episodes from each session for the test set. From the 'Soft' session, we choose the longest (Soft-1) and the shortest (Soft-2) episodes to cover varying trajectory lengths. For the 'Stiff' session, we include Stiff-1, which exhibits outlier wrench magnitudes (627 N in $\mathbf{f_x}$ and 193 Nm in $\mathbf{m_z}$, possibly reflecting the deepest penetration d_z), and the longest episode (Stiff-3). The test set accounts for 33% of the total samples. We use the remaining eight episodes ($\approx 67\%$) for training.

C. Evaluation method

To evaluate the proposed FDN, we compare it against baselines from force estimation and time-series forecasting:

a) Point-to-point estimators: We implement an improved version of MINN [6], as well as RBF neural network [9], and Gaussian process regression (GPR) [10]. These models estimate $\mathbf{W}_t$ from the input vector $\boldsymbol{x}'_t = [\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \ddot{\boldsymbol{q}}_t, \boldsymbol{u}_t]$.

b) Sequence-to-point estimators: We implement LSTM [11] and CNN [12] models, which estimate $\mathbf{W}_t$ from the input sequence $\boldsymbol{x}'_{t_h:t}$.

c) Sequence-to-sequence forecasters: We implement LSTM encoder-decoder (LSTM-ED) [29], Transformer encoder-decoder with generative inference [27], [30], and variants of PatchTST and iTransformer [14], [31]. Since PatchTST and iTransformer were originally designed for endogenous forecasting, where the input and output channels are identical, we incorporate channel-mixing linear projection and modified RevIN described in Section III-B to support our setting. These models forecast $\mathbf{W}_{t+1:t_f}$ from the input history $\boldsymbol{x}'_{t_h:t}$. In addition, we consider PatchTST-Gaussian, which forecasts step- and channel-wise Gaussian parameters of $\mathbf{W}_{t+1:t_f}$ from $\boldsymbol{x}'_{t_h:t}$ as in Eq. 20.

In our experiments, we use $L = 100$ and $T = 100$, corresponding to forecasting one second into the future from one second of history. We use $n = 6$ for models without pretraining. All models are optimized with Adam [32] for 5 epochs using a batch size of 64. We normalize the training datasets to zero mean and unit standard deviation. Hyperparameters are kept as consistent as possible across models, including latent dimension $D = 128$, and patch length $P = 24$. For model-specific hyperparameters, we report their best-performing configurations found. We pretrain FDN on a filtered subset of 11,131 RH20T episodes with valid proprioception and wrench data. For the pretrained FDN, the input normalization statistics from RH20T are reused in the downstream. All experiments are conducted over three runs, and we report the mean values. Further implementation details are available at our GitHub.

For evaluation, we assume a constant time delay t_{delay} that accounts for communication, preprocessing, and inference during real deployment. Under this setting, each model is used to reconstruct the test episodes using a single prediction point from each input sample. For point estimators, the reconstructed episode is shifted backward by t_{delay} and compared with the unshifted ground-truth episode, corresponding to a *delayed zero-order-hold estimate*. For sequence forecasters, we extract the prediction at horizon $t + t_{\text{delay}}$ from each forecasted sequence and use it to reconstruct the episode without shifting. The reconstructed episode is then compared directly with the time-aligned ground-truth episode, corresponding to a *delay-compensated prediction*.

We evaluate band-specific accuracy of the reconstructed trends and residuals obtained by episode-level decomposition using FPF_{low} with $f_c = 1$ Hz. For the high-frequency band, we evaluate transient amplitude reconstruction by comparing the windowed root mean square (RMS) of the residual. Specifically, we compute the RMS of the episode residuals within sliding windows and compute the root mean squared error (RMSE) of the windowed RMS values:

$$\text{wRMSE} = \sqrt{\frac{1}{N-w+1}\sum_{t=1}^{N-w+1}\left(r_t(\hat{y}^{\text{res}}) - r_t(y^{\text{res}})\right)^2} \tag{32}$$

TABLE II
HIGH-FREQUENCY WINDOWED RMS ERRORS OF MODELS.

Metric	wRMSE, High-frequency Windowed RMS Error			
Time Delay	100 ms		1,000 ms	
Force/Torque Unit	[N]	[Nm]	[N]	[Nm]
MINN [6]	23.645	4.801	23.717	4.816
RBF [9]	23.421	4.644	23.454	4.650
GPR [10]	24.262	4.849	24.300	4.861
LSTM [11]	17.941	3.766	18.465	3.837
CNN [12]	24.394	4.989	24.428	4.996
LSTM-ED [29]	22.554	4.574	22.508	4.550
Transformer [27]	24.019	4.887	23.958	4.870
PatchTST [14]	22.274	4.565	23.059	4.720
PatchTST-Gaussian	15.336	3.377	15.658	3.412
iTransformer [31]	21.992	4.515	23.135	4.742
FDN (Scratch)	<u>12.912</u>	**2.593**	<u>14.355</u>	**2.903**
FDN (Pretrained)	**11.876**	<u>2.621</u>	**13.798**	<u>2.997</u>

where the window RMS function r_t is defined as:

$$r_t(x) = \sqrt{\frac{1}{w}\sum_{i=t}^{t+w-1}(x_i^2)} \tag{33}$$

We set the sliding window size $w = 10$, corresponding to a 0.1 s window. For the low-frequency band, we compute pointwise RMSE of the trends:

$$\text{pRMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i^{\text{trend}} - \hat{y}_i^{\text{trend}})^2} \tag{34}$$

To additionally evaluate predictions over the full frequency band, we measure the continuously ranked probability score (CRPS), a proper scoring rule widely used for evaluating probabilistic forecasts:

$$\text{CRPS} = \frac{1}{N}\sum_{i=1}^{N}\int_{-\infty}^{\infty}\left(F_i(x) - \mathbf{1}_{\{x \geq y_i\}}\right)^2 dx \tag{35}$$

where F_i is the predicted cumulative distribution function (CDF) and $\mathbf{1}_{\{x \geq y_i\}}$ is the step CDF at observation y_i. The CRPS reduces to the mean absolute error (MAE) for deterministic models, since $F_i(x) = \mathbf{1}_{\{x \geq \hat{y}_i\}}$.

For the probabilistic models (GPR, PatchTST-Gaussian, and FDN), episode-level trends are obtained by decomposing their predictive mean. For FDN, we use $\hat{\mathbf{W}}^{\text{trend}} + \hat{\boldsymbol{\mu}}^{\text{res}}$ as its predictive mean. In addition, FDN and PatchTST-Gaussian directly parameterize the wrench distribution, whereas GPR models posterior uncertainty. Therefore, for FDN and PatchTST-Gaussian, we replace the energy x_i^2 in Eq. 33 with its expectation $\mathbb{E}[x_i^2] = \mu_i^2 + \sigma_i^2$. Here, μ_i denotes the residual of predictive mean, and σ is the predicted standard deviation. For GPR, all metrics are computed from its predictive mean only.

D. Comparative analysis

Table II and III report high- and low-frequency band-specific metrics. Models that explicitly parameterize the wrench distribution show clear improvements in wRMSE,

TABLE III
LOW-FREQUENCY POINTWISE RMSE OF MODELS.

Metric	pRMSE, Low-frequency Pointwise RMSE			
Time Delay	100 ms		1,000 ms	
Force/Torque Unit	[N]	[Nm]	[N]	[Nm]
MINN [6]	12.230	3.612	12.274	3.623
RBF [9]	10.455	3.076	10.432	3.067
GPR [10]	9.362	**2.285**	**9.362**	**2.281**
LSTM [11]	13.040	3.289	13.189	3.300
CNN [12]	10.322	2.485	10.766	2.511
LSTM-ED [29]	11.873	3.199	11.889	3.231
Transformer [27]	10.939	2.954	11.009	3.023
PatchTST [14]	9.898	2.545	10.223	2.578
PatchTST-Gaussian	13.562	4.358	13.563	4.352
iTransformer [31]	9.994	2.587	10.326	2.612
FDN (Scratch)	**9.102**	3.825	9.411	3.830
FDN (Pretrained)	10.448	3.045	10.659	3.055

TABLE IV
CONTINUOUSLY RANKED PROBABILITY SCORES OF MODELS.

Metric	CRPS			
Time Delay	100 ms		1,000 ms	
Force/Torque Unit	[N]	[Nm]	[N]	[Nm]
MINN [6]	12.780	3.846	12.803	3.850
RBF [9]	12.324	3.800	12.337	3.800
GPR [10]	11.855	3.461	11.884	3.469
LSTM [11]	14.410	4.278	14.643	4.317
CNN [12]	12.531	3.668	12.608	3.680
LSTM-ED [29]	13.156	4.007	13.205	4.006
Transformer [27]	12.648	3.843	12.715	3.883
PatchTST [14]	12.015	3.751	12.183	3.769
PatchTST-Gaussian	9.596	3.238	9.617	3.235
iTransformer [31]	11.968	3.778	12.101	3.773
FDN (Scratch)	**8.891**	3.213	**8.964**	3.219
FDN (Pretrained)	9.129	**2.931**	9.224	**2.952**

indicating the effectiveness of distribution parameterization in the high-frequency band. In particular, FDN demonstrates the best performance in the high-frequency band, reducing wRMSE by up to 50% relative to the baselines while maintaining competitive pRMSE in the low-frequency band. In contrast, most baselines exhibit a clear imbalance between the two band-specific metrics. They generally perform well in the low-frequency band, but their accuracy degrades substantially in the high-frequency band. PatchTST-Gaussian improves high-frequency reconstruction by parameterizing the full-band wrench distribution. However, this gain accompanies increased error in the low-frequency band, implying that distribution parameterization alone is insufficient to simultaneously improve both band-specific metrics. In contrast, FDN effectively mitigates this imbalance with decomposition-based asymmetric modeling and shows consistent competitiveness across bands. Table IV also shows that FDN achieves the lowest CRPS overall across models and time delays, illustrating its superior full-band performance.

Fig. 6 further visualizes these results using the pretrained FDN and two representative baselines, GPR and PatchTST-Gaussian. Although GPR is one of our strongest baselines, it shows limited ability to reconstruct high-frequency vibrations, particularly in stiff episodes where wrench magnitudes are larger. PatchTST-Gaussian reconstructs the high-frequency amplitudes through its learned full-band distribution. Nevertheless, its distribution does not accurately capture contact transients. In contrast, FDN successfully captures local high-frequency fluctuations and peaks, and overall low-frequency trends, which highlights its strong capability for contact- and vibration-rich wrench prediction.

Note that adopting a forecasting formulation does not necessarily improve prediction accuracy over estimator baselines. Delayed zero-order-hold estimates remain competitive with the forecasting baselines even under $t_{\mathrm{delay}} = 1,000$ ms. However, the formulation allows us to adopt advanced time-series backbones and model the sequential structure of wrench trajectories, which motivates decomposition-based forecasting, frequency filtering, and transfer learning in FDN. Pretraining also yields additional gains across metrics, as detailed in the following section.

TABLE V
ARCHITECTURAL ABLATION RESULTS.

Metric	HF wRMSE	LF pRMSE	CRPS
w/o FEF	0.5090 (+3.6%)	0.5687 (+6.7%)	0.5270 (+3.0%)
w/o FEF-W	0.5092 (+3.6%)	0.5719 (+7.3%)	0.5273 (+3.0%)
w/o FEF-MoE	0.5070 (+3.2%)	0.5651 (+6.0%)	0.5241 (+2.4%)
w/o FPF	**0.4913** (-0.0%)	0.5367 (+0.7%)	0.5135 (+0.4%)
w/o ModSpec	0.5426 (+10%)	0.6064 (+14%)	0.5526 (+8.0%)
w/o TrdHead	0.6106 (+24%)	0.6314 (**+18%**)	**0.5066** (-1.0%)
w/o ResHead	0.7490 (**+52%**)	0.5367 (+0.7%)	0.6672 (**+30%**)
FDN	0.4915	**0.5329**	0.5117

E. Ablation studies

Table V shows architectural ablation results of FDN. We remove the frequency enhancement filter (w/o FEF), its expert weighting while retaining multiple filters (w/o FEF-W), its MoE design by reducing to a single unweighted filter (w/o FEF-MoE), the frequency-pass filters in Eq. 21 (w/o FPF), the modality-specific encoders by replacing them with a single shared encoder for $\boldsymbol{x}^{\delta}_{t_h:t}$ (w/o ModSpec), and each prediction head (w/o TrdHead and w/o ResHead). When ablating a head, we retain frequency-pass filtering to the remaining output, i.e., $\tilde{\boldsymbol{\mu}}^{\mathrm{res}}$ or $\tilde{\mathbf{W}}^{\mathrm{trend}}$, using $\mathrm{FPF}_{\mathrm{low}}$ with $f_c = f_c^{\mathrm{dn}} = 15$ Hz. All values are reported on a normalized scale to reduce channel-wise magnitude effects.

The results indicate that the residual head is the main contributor to modeling the high-frequency band. Removing it, which reduces FDN to a full-band pointwise regressor, increases wRMSE by 52%. Removing the trend head, which corresponds to parameterizing a full-band distribution as in PatchTST-Gaussian, also degrades both wRMSE and pRMSE by around 20%, despite a slight reduction in CRPS. Consistent with the previous section, these results show that relying solely on pointwise regression or distribution parameterization over the full band is ineffective. Rather, applying asymmetric modeling across frequency bands facilitates the band-balanced performance as in FDN. Replacing the modality-specific encoders with a single encoder also degrades all

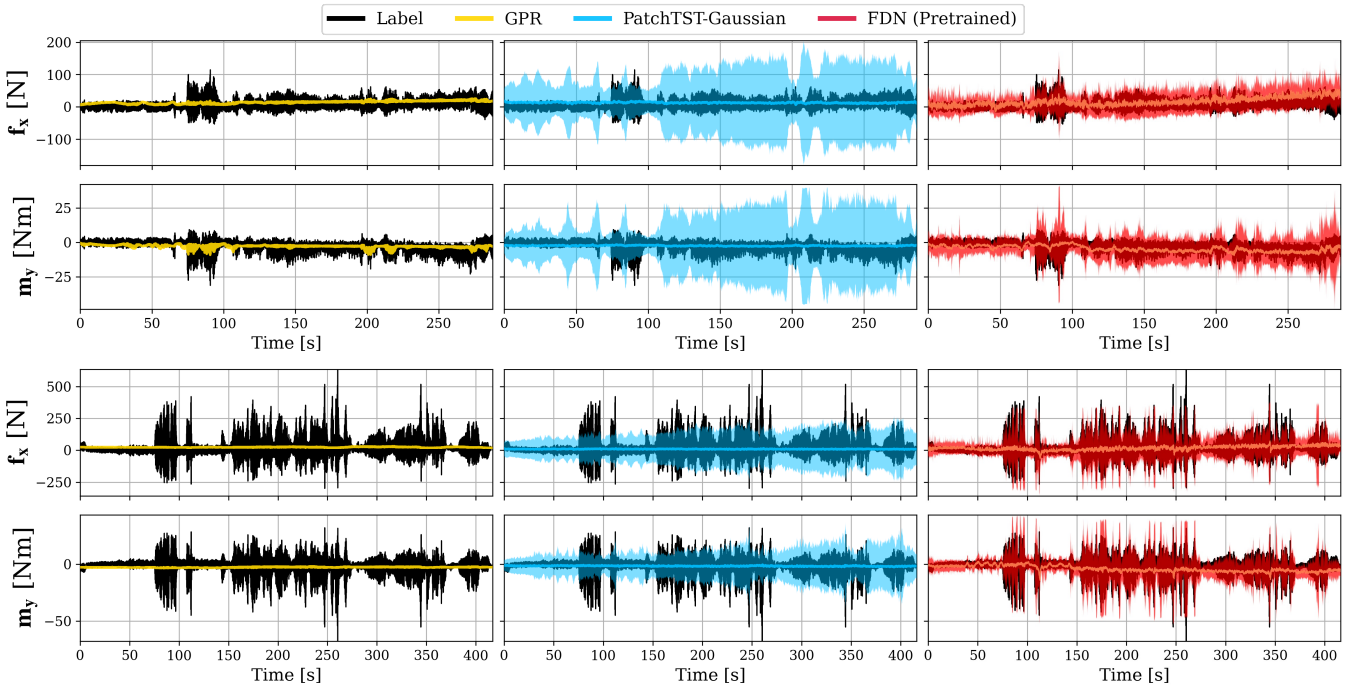


Fig. 6. Test episode reconstructions with $t_{\text{delay}} = 100$ ms. We visualize the $\mathbf{f_x}$ and $\mathbf{m_y}$, which are representative channels in our experimental setting described in Section IV-A. The upper two rows correspond to the 'Soft-1' episode, and the lower rows correspond to the 'Stiff-1' episode. Colored areas illustrate the prediction interval defined by $\mu \pm 3\sigma$.

metrics by around 10%, suggesting that the heterogeneous input modalities are better handled with separate encoders.

For the frequency-aware layers, FEF provides consistent but moderate benefit, as removing it increases wRMSE by 3.6% and pRMSE by 6.7%. Removing either the expert weighting or the MoE structure from the FEF also yields a similar level of degradation, indicating that its benefit is not retained without its complete design. Meanwhile, the quantitative effect of FPF is small, causing less than 1% change across all metrics. Thus, we view it as a design for explicitly imposing our frequency-band prior derived from the ground-truth decomposition in Eq. 11, rather than a primary source of improvement.

F. Transfer learning analysis

Table VI summarizes transfer learning results of our pretraining under a fixed budget of 100K training iterations. Here, we vary the data utilization ratios from 20% to 100% of the 11,131 pretraining episodes. We also assess the effect of input formulations on transfer by comparing absolute position inputs ('A') with the default relative position inputs ('R') in Eq. 17. For the absolute position formulation, we omit the initial position encoder $\text{Enc}_{\boldsymbol{q}_0^e}$.

Under the relative position formulation, fine-tuning generally outperforms linear probing, indicating that end-to-end adaptation is required to realize transfer gains. In particular, fine-tuning with relative positions improves the aggregated normalized-scale pRMSE and CRPS by up to 8% and 4% over training from scratch, while slightly degrading wRMSE.

TABLE VI
TRANSFER LEARNING RESULTS ON NORMALIZED SCALE. LP REFERS TO LINEAR PROBING, AND FT REFERS TO FINE-TUNING. A AND R DENOTE ABSOLUTE AND RELATIVE POSITION INPUTS. 0% CORRESPONDS TO TRAINING FROM SCRATCH.

Pretraining Data Util.		0%	20%	40%	60%	80%	100%
Num. Episodes		-	2,226	4,452	6,678	8,904	11,131
Effective Epochs		-	4.93	2.52	1.66	1.25	1.00
HF wRMSE	LP(A)	0.519	0.547	0.546	0.628	0.605	0.604
	FT(A)		0.534	0.526	0.544	0.538	0.537
	LP(R)	**0.492**	0.554	0.568	0.556	0.572	0.560
	FT(R)		0.499	0.498	0.496	0.506	0.506
LF pRMSE	LP(A)	0.511	0.528	0.530	0.580	0.574	0.568
	FT(A)		0.504	0.507	0.572	0.573	0.549
	LP(R)	0.533	0.541	0.539	0.552	0.552	0.548
	FT(R)		0.494	0.494	**0.490**	0.504	0.499
CRPS	LP(A)	0.510	0.508	0.507	0.542	0.540	0.535
	FT(A)		0.501	0.500	0.542	0.542	0.532
	LP(R)	0.512	0.512	0.511	0.519	0.518	0.516
	FT(R)		0.491	0.492	**0.490**	0.499	0.496

Although this suggests more evident transfer gain in the low-frequency band, Table II shows that transfer gain also exists in high-frequency force, where wRMSE decreases by 4 to 8%. Notably, Table III shows the clearest gain in low-frequency torque where pRMSE decreases by 21%.

We analyze these results with the dataset properties summarized in Table VII. RH20T is strongly low-frequency dominant, whereas the downstream hydraulic wrench is dominated by high-frequency content. This spectral mismatch reflects the physical differences between the two settings, mainly in

TABLE VII
PROPERTIES OF PRETRAINING AND DOWNSTREAM DATASETS, INCLUDING BAND ENERGY RATIOS OF THE WRENCH. 'LF' AND 'HF' INDICATE BANDS IN $f \leq 1$ HZ AND $f > 1$ HZ, RESPECTIVELY.

Datasets	Pretraining (RH20T [16])	Downstream
HF $\mathbf{W}$ Energy (%)	9.843%	**84.930%**
LF $\mathbf{W}$ Energy (%)	**90.157%**	15.070%
Task Domain	Contact-rich Manipulation	Block Grinding
Actuation	Electric	Hydraulic
Actuation Signal	Motor Torque	Diff. Pressure
End-effector	Parallel Gripper	Rotary Grinder
Robot Embodiment	6- and 7-DoF Arms	6-DoF Arm

the task domain and actuation. Nevertheless, the two datasets are likely to retain shared structures since both measure contact-rich wrist wrenches of arm manipulators. Our transfer results imply that such structures exist between datasets in a transferable form, and are more apparent in the low-frequency domain in our case. Furthermore, the masked pretraining in Section III-F suggests that the transferred representation is unlikely to depend on the joint actuation signals or actuator torques. We thus infer that the transferred representation plausibly encodes coarse proprioception-to-wrench coupling in arm manipulators based on Eq. 5, and temporal evolution patterns of wrenches shared across datasets. The high-frequency dynamics of the downstream wrench appear to be more domain-specific to the hydraulic excavation setting, consistent with the band energy mismatch and the weaker transfer gain in wRMSE. Meanwhile, the absolute position formulation neither provides consistent transfer gains nor outperforms the relative position formulation, indicating that aligning episode-wise offsets in proprioception improves transferability in our setting.

Across utilization ratios, we find the best transfer performance at 60% utilization under the relative position formulation. Since the effective number of epochs decreases as the amount of pretraining data increases under the fixed iterations, this result should be interpreted as a budget-dependent optimum.

V. CONCLUSIONS

In this work, a Frequency-aware Decomposition Network (FDN) was proposed for sensorless wrench forecasting on a vibration-rich hydraulic manipulator. In our robotic grinding excavation, the target wrench exhibited substantial high-frequency vibrations that are task-critical and difficult to predict. To address this, the proposed model combined decomposition-based probabilistic modeling, frequency-aware layers, and large-scale proprioception-to-wrench pretraining. Under a delayed estimation setting, FDN outperformed baselines from both robotic wrench estimation and time-series forecasting, and effectively mitigated the imbalance between low- and high-frequency band accuracies evident in most baselines. Transfer learning provided additional gain mainly in the low-frequency domain, suggesting the presence of transferable proprioception-to-wrench structure across heterogeneous robot platforms and tasks.

Our results suggest that the proposed framework can be useful for wrench-based robotic applications, particularly for robots involving vibration-rich tasks, platforms, and environments. Nevertheless, the present study has several limitations. First, the evaluation is limited to a single hydraulic manipulator and excavation setting, and broader generalization across contact objects, robot platforms, and tasks remains to be validated. Second, pretraining and transfer learning warrant deeper investigation. Further analyses of scaling behavior and representation learning strategies, as well as additional utilities such as improved inference robustness or few-shot generalization, would provide a more rigorous understanding of their effects on wrench estimation and forecasting. Lastly, investigating the use of forecasting models in real-world deployment would further clarify their practical value in delay-aware wrench estimation and predictive control settings.

REFERENCES

[1] G. Tholey, J. P. Desai, and A. E. Castellanos, "Force feedback plays a significant role in minimally invasive surgery: results and analysis," *Annals of surgery*, vol. 241, no. 1, pp. 102–109, 2005.

[2] C. González, J. E. Solanes, A. Munoz, L. Gracia, V. Girbés-Juan, and J. Tornero, "Advanced teleoperation and control system for industrial robots based on augmented virtuality and haptic feedback," *Journal of Manufacturing Systems*, vol. 59, pp. 283–298, 2021.

[3] L. Villani and J. De Schutter, "Force control," in *Springer handbook of robotics*. Springer, 2016, pp. 195–220.

[4] S. Stepputtis, M. Bandari, S. Schaal, and H. B. Amor, "A system for imitation learning of contact-rich bimanual manipulation policies," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 810–11 817.

[5] M. Y. Cao, S. Laws, and F. R. y Baena, "Six-axis force/torque sensors for robotics applications: A review," *IEEE Sensors Journal*, vol. 21, no. 24, pp. 27 238–27 251, 2021.

[6] A. C. Smith, F. Mobasser, and K. Hashtrudi-Zaad, "Neural-network-based contact force observers for haptic applications," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1163–1175, 2006.

[7] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robotics and computer-integrated manufacturing*, vol. 26, no. 5, pp. 414–419, 2010.

[8] R. E. Ellis and S. L. Ricker, "Two numerical issues in simulating constrained robot dynamics," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 1, pp. 19–27, 2002.

[9] Z. Chen, F. Huang, W. Sun, J. Gu, and B. Yao, "Rbf-neural-network-based adaptive robust control for nonlinear bilateral teleoperation manipulators with uncertainty and time delay," *Ieee/Asme Transactions on Mechatronics*, vol. 25, no. 2, pp. 906–918, 2019.

[10] A. Dong, Z. Du, and Z. Yan, "A sensorless interaction forces estimator for bilateral teleoperation system based on online sparse gaussian process regression," *Mechanism and Machine Theory*, vol. 143, p. 103620, 2020.

[11] S. Kružić, J. Musić, R. Kamnik, and V. Papić, "End-effector force and joint torque estimation of a 7-dof robotic manipulator using deep learning," *Electronics*, vol. 10, no. 23, p. 2963, 2021.

[12] M.-Z. Pan, J.-A. Li, Z. Li, K. Liang, T.-C. Su, K. Liang, and G.-B. Bian, "A graph robot network for force observer of teleoperation systems," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 1, pp. 530–540, 2024.

[13] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

[14] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

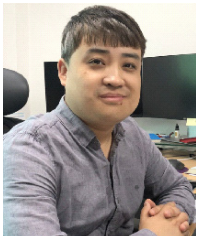
[15] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[16] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 653–660.
[17] D. Xu, L. Yin, and J. Wang, "Grinding force estimation and control of grinding robot with variable impedance control strategy," *The International Journal of Advanced Manufacturing Technology*, vol. 137, no. 3, pp. 2011–2027, 2025.
[18] J. Hu, Z. Zhou, G. Xia, Y. Dai, J. Zhang, G. Yang, X. Han, J. Jiang, and Y. Liu, "Accurate milling force estimation and surgical state recognition in robot-assisted laminectomy," *Measurement*, vol. 253, p. 117673, 2025.
[19] Y. Dong, T. Ren, K. Hu, D. Wu, and K. Chen, "Contact force detection and control for robotic polishing based on joint torque sensors," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 5, pp. 2745–2756, 2020.
[20] J. T. Jose, J. Das, and S. K. Mishra, "Dynamic improvement of hydraulic excavator using pressure feedback and gain scheduled model predictive control," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 18 526–18 534, 2021.
[21] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 5301–5310.
[22] H. Lee, S. Han, H.-S. Choi, and J.-G. Kim, "cnn-dp: Composite neural network with differential propagation for impulsive nonlinear dynamics," *Journal of Computational Physics*, vol. 496, p. 112578, 2024.
[23] J. Jung, J. Lee, and K. Huh, "Robust contact force estimation for robot manipulators in three-dimensional space," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 220, no. 9, pp. 1317–1327, 2006.
[24] Y. Cheng, Y. Li, X. Liu, and Y. Cai, "Mechanism-based structured deep neural network for cutting force forecasting using cnc inherent monitoring signals," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2235–2245, 2021.
[25] C. Ni, J. Yang, and H. Ding, "Unsupervised domain adversarial adaptive regression network for cutting force prediction at varying spindle speeds," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 1, pp. 252–263, 2024.
[26] S. Butterworth *et al.*, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[28] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International conference on learning representations*, 2021.
[29] I.-F. Kao, Y. Zhou, L.-C. Chang, and F.-J. Chang, "Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting," *Journal of Hydrology*, vol. 583, p. 124631, 2020.
[30] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.
[31] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.
[32] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Hyeonbeen Lee received the B.Eng. and M.Eng. degrees in mechanical engineering from Kyung Hee University, Seoul, South Korea, in 2022 and 2024, respectively. He is currently an incoming Ph.D. student at Virginia Tech, Blacksburg, VA, USA. His research interests include contact-rich manipulation, physics-aware machine learning, nonlinear dynamics, and robot learning.

Min-Jae Jung received the B.Eng. degree in mechanical engineering from Kyung Hee University, Seoul, South Korea, in 2025, where he is currently pursuing the M.Eng. degree. His research focuses on robot manipulation using force/torque sensing and machine learning, with an emphasis on contact-rich assembly, time-series modeling, and physics-informed learning for multibody dynamics systems.

Tae-Kyeong Yeu received the B.Eng. and M.Eng. degrees in mechanical engineering from Pukyong National University, Busan, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree in information systems engineering from Kumamoto University, Kumamoto, Japan, in 2003. He is currently a principal researcher at Korea Research Institute of Ships and Ocean Engineering (KRISO), Daejeon, South Korea. His research interests include the design and control of underwater robots and cyber-physical systems (CPS).



Jong-Boo Han received the B.Eng., M.Eng., and Ph.D. degrees in mechatronics engineering from Chungnam National University, Daejeon, South Korea, in 2009, 2011, and 2018, respectively. He is currently a senior researcher at Korea Research Institute of Ships and Ocean Engineering (KRISO), Daejeon, South Korea. His research interests include multibody dynamics modeling and real-time physics engines.

Daegil Park received the B.S. degree in mechanical engineering from Seoul National University of Science and Technology, South Korea, in 2011, and the Ph.D. degree in mechanical engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2016. He is currently a senior researcher at Korea Research Institute of Ships and Ocean Engineering (KRISO), Daejeon, South Korea, and an Associate Professor at the University of Science and Technology (UST), Daejeon, South Korea. His research interests include underwater robots, autonomy, and control of robot-environment interactions.

Jin-Gyun Kim received the B.S. and M.S. degrees in civil and environmental engineering from Korea University, Seoul, South Korea, in 2008 and 2010, respectively, and the Ph.D. degree in ocean systems engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014. He is currently an Associate Professor and Vice Dean of the College of Engineering at Kyung Hee University, Seoul, South Korea, and a Visiting Professor at the University of Auckland, Auckland, New Zealand. His research interests include modeling and simulation of dynamics, vibrations, and multiphysics.